COMBINED FEATURE-LEVEL VIDEO INDEXING USING BLOCK-BASED MOTION ESTIMATION

Harish Bhaskar
Department of Computer Engineering
Khalifa University, Sharjah Campus
U.A.E.
harishbhasky@gmail.com

Lyudmila Mihaylova
Department of Communication Systems
Lancaster University
U.K.
mila.mihaylova@lancs.ac.uk

Abstract – *We describe a method for attaching content-based labels to video data using a weighted combination of low-level features (such as colour, texture, motion, etc.) estimated during motion analysis. Every frame of a video sequence is modeled using a fixed set of low-level feature attributes together with a set of corresponding weights using a block-based motion estimation technique. Indexing a new video involves an alternative scheme in which the weights of the features are first estimated and then classification is performed to determine the label corresponding to the video. A hierarchical architecture of increasingly complexity is used to achieve robust indexing of new videos. We explore the effect of different model parameters on performance and prove that the proposed method is effective using publicly available datasets.*

Keywords: Tracking, filtering, estimation, fuzzy logic, resource management.

1 Introduction

Video indexing involves attaching content or appearance-type labels to video in order to provide efficient, reliable and classified access to the data [5, 22]. Video indexing is a central problem in the structured organization of video data that will allow efficient retrieval, browse and manipulation. However, these tasks are particularly complicated when performed without appropriate human intervention [10]. A number of different approaches for video indexing have been proposed, mostly encapsulating variations in either low-level (physical) or high-level (semantic) feature attributes. Though high-level indices provide elaborate description of video content, they are often contaminated by semantic inconsistencies and are generally suitable for dealing with small quantities of video and providing access to already annotated video data. On the other hand, low-level video indexing based on feature attributes such as colour [4], texture [16], motion [17], etc. provide basis for video classification. The general idea behind these methods is to extract features from the video, organize them based on distances and use some form of similarity matching for video classification [3].

In this paper we investigate a method of combined feature-level motion estimation for accurate and robust video indexing. The central idea behind the proposed method is to estimate some average feature-level characteristics of a video using a block-based motion estimation scheme. In accordance with the features selected for matching the blocks i.e. grey-level, texture, colour, motion, etc. and to the weights assigned to these features, we obtain performance levels in terms of the fitness between each frame and motion estimated frame by means of a block matching procedure. The weights leading to the best average matches between all the pairs of actual and motion estimated frames along a sequence are considered as the descriptors of the video. These descriptors are then used for assigning videos to different classes using an classification algorithm. We denote this method a *combined feature-level video indexing* (CFVI) and demonstrate it on a various publicly available datasets including CAVIAR [1], CMU [13], PETS 2001 [2] and our own collection of People, Traffic and Under-water videos.

1.1 Related Work

The goal of video indexing and retrieval systems is to model and extract effective features describing the visual input being indexed and thus use these extracted features to search and match the query video based on a suitable similarity rank. The chosen feature descriptors can be either low-level (primitive), high-level (semantic) or more popularly a combination of the two [5].

Under the high-level (semantic) video indexing framework, information on the high-level ontological categories such as objects, actions, time, abstract are often used as descriptors for indexing and retrieval. This video abstract is in a form of a short sequence of images, extracted from longer sequences in a way that it preserves the underlying meaning/message of the original video [15, 17]. The obvious complexity in extracting such an abstract is in determining which frames best represent the contents of the original video. A general approach is to segment a video in shots and then select a key frame to represent each shot. Several different approaches have been proposed to summarize and further index videos based on this principle. In [15], the authors aim to generate trailers for movies that result in a precise video abstract, however, this may not work with video documentaries and libraries where the video content is exhaustive and elaborate. In contrast, explicit detection of shot changes have been more widely used for video indexing [14]. A typical approach for semantic video indexing is to use supervised learning techniques such as the graphical models and support vector machines [16, 24] or rarely unsupervised schemes to extract repeating feature patterns such as visual context, camera motion and audio context [11, 22]. In addition to conventional techniques such as the edit effect detection algorithm outlined in [18] and extraction of temporal aspects of video data as in [17], much research efforts have been spent in extracting temporal coherence among shots for video indexing [19, 25]. A majority of the techniques introduced above, capture and preserve high level description of visual information by modeling description of some event or concept [24], object motion information [9] and influences of capture devices [17]. In order to handle the constraints in time, storage and efficiency overhead of delivering such high-level descriptions, a large number of high level techniques either directly or indirectly depend on low-level (primitive) features.

Low-level features that are used to represent video data have conventionally been the same used for images, generally including the temporal aspects of video data (in the form of motion primitives). As previously mentioned low-level features form the lowest level of abstraction that models descriptions of the raw video data that is used as input for the higher-levels. In [16], region level feature extraction is performed by combining linearized Hue, Saturation and Value (HSV) colour histograms with gray-level co-occurrence matrix type texture and affine motion features for accurate video indexing. In a similar study by [8], Luminance L and Chrominance components U and V (LUV) colour, coarseness, contrast and orientation textures, shape and motion features have been combined for accurate video indexing. However, these features are extracted after objects of interest are segmented from the video using a high-level spatio-temporal segmentation procedure. A novel technique of multi-resolution analysis of colour histograms using Hausdroff similarity criteria addressing the scalability and efficiency issues of video indexing systems is proposed in [7]. Likewise [26] integrate motion features computed using perceived motion energy spectrum (PMES) with camera motion descriptor based on normalized dominant direction histogram and quantized colour primitives for classifying video data using support vector machines. With a total of 40 features the authors have demonstrated that the model can capture fundamental characteristics of the video data and can achieve high levels of generalization.

Though so much research efforts have been spent on developing state-of-the-art systems for video indexing, several open issues still remain. Though recent literature suggests the use of Principle Component Analysis (PCA) for dimensionality reduction during indexing [12], the problem of high dimensional feature space still remains fully unresolved. In addition to the issues of lack of generality [21] and not having benchmark data and performance evaluation criteria, the gap between composing low-level and high-level features for extracting semantic features is significant [5].

2 Contributions and Structure

The method proposed in this paper combines weighted feature descriptors estimated using block matching with a classification strategy for video indexing. One novelty of the method is that the estimation of weights for features describing a video is integrated into block matching based motion estimation. The technique automatically categorizes videos quantitatively by the relative proportion of feature attributes that are present in them. Furthermore, the method applies a hierarchical implementation for robustness: in the first step, we determine the weights of a salient subset of features using block-based motion estimation; in the second step we refine the current estimates of weights using the combined weights from the previous step by comparing blocks that share similar motion characteristics and also estimate new weights corresponding to a larger set of less salient features. Our results suggest that using such a hierarchical architecture of increasing complexity can significantly improve the performance when compared to a single-level model.

The paper is organized as follows. We begin by describing our CFVI model, including the motion estimation technique in Section 3. We then conduct experiments on various publicly available dataset in Section 4 that investigate the effect of: the hierarchical implementation, the measure of performance, the choice of the search strategy used within motion estimation when compared to other developed techniques. In Section 5, we present some conclusive remarks and future directions of work.

3 Proposed Method

In our proposed technique, we formulate the video indexing problem using low-level (primitive) features and their corresponding weights that are estimated during motion estimation using a block-based approach. We represent the model as a set of N primitive features and their corresponding weights, $\Re = \{\ \aleph_i = (f_i, w_i)\ \}$. Given a query video V, our aim is then to find the optimal set, $\Re^*$ that maximizes the performance metric Ω. The distribution of weights of corresponding features of the query video $\mathbf{V}$ is then subjected to classification to determine the class label (known from already available ground truth). We summarize this approach as follows:

1. Initialize the feature set f_i^0 and their corresponding weights, $w_i^0 = \frac{u_i}{\sum_{j=1}^{i} u_j}$; where u_i is a random number generated between [0,1].
2. For image frames I at t = 1, . . ., T-1:
   - For $\ell = 1, ..., K$ iterations:
     - Select the best candidate target frame, $I^*_{t+1} \leftarrow m(I_t, I_{t+1})$, where m is the fitness function of weighted block based motion estimation.
     - Measure the performance of motion estimation using known metric d to obtain $\Omega_\ell = d(I^*_{t+1}, I_{t+1})$.
     - Reinitialize $w_i^{0,\ell} = \frac{u_i}{\sum_{j=1}^{i} u_j}$; where u_i is a random number generated between [0,1].
   - Select update weights $w_i^{t,\ell} = \text{argmax}_\ell \Omega$.
3. Estimate the average weights $\overline{w}_i = \frac{w_i^{t=1:T-1}}{T-1}$ as the feature descriptors of the video.
4. Perform classification on the descriptors to obtain relevant class label.

3.1 Motion Estimation

The first step involves finding the best candidate target frame, I^*_{t+1} from the source frame I_t, using motion vectors computed from a weighted block based motion estimation framework. The proposed weighted block based motion estimation technique extends the block matching method suggested in [6]. The block matching strategy in [6] is implemented using an affine based genetic algorithm search scheme for accurate motion estimation. In this paper we extend that genetic algorithm search mechanism such that it accommodates matching weighted features.

Let us assume that b_t represents the block extracted from the source image frame after quad-tree decomposition and $\overline{b}_{t+1}$ be the affine matched block extracted from the target frame. The genetic algorithm is an evolving iterative procedure that aims to minimize an fitness function. The fitness function for the proposed weighted genetic algorithm search is as follows:

$$Minimize\ m(.) \quad = \quad argmin \sum_{i=1}^{N} w_{f_i} * m_{f_i} \qquad (1)$$

where m_{f_i} measure the cumulative difference between the features (pixel-wise or regions-wise) of block b_t and $\overline{b}_{t+1}$,

$$m_{f_i} \quad = \quad |b_t^{f_i} \ominus \overline{b}_{t+1}^{f_i}| \qquad (2)$$

where $\ominus$ is a difference operator. For example, if grey level intensity feature is used, then m_{f_i} will be cumulative sum of mean absolute difference between pixels of the block $b_t^{f_i}$ and $\overline{b}_{t+1}^{f_i}$.

In our genetic algorithm the population is initialized with chromosomes that are encoded as a vector containing the displacements in x and y directions and a set of affine parameters that encompasses rotational, shear, scale and squeeze changes: $(\partial x, \partial y, a_{11}, a_{12}, a_{21}, a_{22})$. For the purposes of evolution we perform the cross-over step based on a single point swap of a random gene between successive chromosomes and the mutation step where we replace all the genes of a particular chromosome using scaled values from a uniform distribution. The algorithm is set to terminate either when the minimum of the fitness function or a maximum number of generations is reached (whichever is earlier). We represent the motion vector between the source image frame I_t and target image frame I_{t+1} as the displacement of a generic point at location (x, y) from t to $t + 1$, we reconstruct the target frame using these vectors to form I^*_{t+1}.

3.2 Performance Metric

Having obtained the reconstruction of the target frame, we focus our attention in measuring the performance of motion estimation based on prediction errors. We measure the quality of motion estimation using Peak signal to Noise Ratio (PSNR), which the most classic metric of evaluating motion estimation. PSNR for a gray scale image is defined as:

$$d_{PSNR} \quad = \quad 10\ log_{10} \left[\frac{255^2}{\frac{1}{HW} \sum_H \sum_W \left\| I_{t+1} - I^*_{t+1} \right\|^2} \right] \qquad (3)$$

where, (H, W) refers to the height and the width of the image frames. PSNR values generally range between 20dB and 40dB; higher values of PSNR indicate better quality of motion estimation.

In addition to the classic measures of quality, we also measure the structural similarity between the two images (SSIM index) comparing local patterns of pixel intensities that have been normalized in luminance and contrast [23].

$$d_{SSIM} = \frac{(2\mu_{I_{t+1}}\mu_{I^*_{t+1}} + c_1)(2cov_{I_{t+1},I^*_{t+1}} + c_2)}{(\mu^2_{I_{t+1}} + \mu^2_{I^*_{t+1}} + c_1)(\sigma^2_{I_{t+1}} + \sigma^2_{I^*_{t+1}} + c_2)} \tag{4}$$

where $c_1 = (k_1 L)^2$, $c_2 = (k2L)^2$, L is the dynamic range of the pixel-values, k_1 and k_2 are two constants equal to 0.01 and 0:03 respectively and μ, cov and σ are the mean, the covariance and the standard deviation functions, respectively.

3.3 Hierarchical Implementation

We extend the proposed model by applying the block matching algorithm in a hierarchical fashion by adapting the quad-tree decomposition methodology of splitting an image into blocks of regions. The quad-tree decomposition scheme progresses recursively dividing the source image frame I_t into 4 equal blocks (regions in the image) until no further splitting is needed. During every cycle of split, the genetic algorithm based search strategy as described in section 3.1 is engaged to accurately match each block to the target frame and thus motion vectors are determined. In our implementation, during the first cycle of decomposition, we determine the weights of only a salient subset of features (up to 3); in the second cycle we refine the current estimated weights of individual blocks using the combined estimate from the previous step by comparing blocks that share similar motion characteristics and also estimate new weights corresponding to a slightly larger set of less salient features (increased by 1 at every level) and so on until no further splitting is needed (maximum sized block of 4x4 pixels). Importantly, search parameters are re-learned at every level so that the nearly correct values are employed rather inefficiently assuming (random) constant values for all levels.

4 Results and Analysis

In this section we perform systematic experiments evaluating the proposed model for its accuracy and robustness. We compare the proposed motion estimation scheme with conventional mechanisms using the standard performance metrics detailed in section 3.2 and the overall performance using the rate of true classification and misclassification. To demonstrate the performance we use a combined dataset that contains a total of 400 short video clips from various different sources including the CAVIAR [1], CMU [13], PETS 2001 [2] and our own collection of people, traffic, maritime surveillance and under-water videos. Each video clip is edited to a length 15 seconds containing approximately 375 frames at a scaled resolution of 320x320. We have manually annotated each of these videos with 1 of 6 possible class labels: 1-indoor surveillance, 2-outdoor surveillance, 3-sports, 4-under-water, 5-wildlife and 6-coastal surveillance. For all our experiments, we use a pool of texture, colour and grey-intensity features ranging from the most salient to the least salient including: correlation, hue, grey-level intensity, contrast, saturation, gradient, energy, brightness and homogeneity. Table 1, summarizes the performance of the CFVI system. Due to space constraint, in Figure 1, we just present the 3 fold results of cross validation of 30 randomly selected test video clips represented using black '*'s. Using the nearest neighbor approach we spatially link the test video clips to their nearest class label. Though it is possible to extend our model using other classification strategies, we restrict ourselves to the nearest neighbor method because of its simplicity. Also, we presume that other classification methods can only improve our results and not otherwise. The results thus obtained have demonstrated 94% classification rate for the chosen features (grey-intensity, hue and correlation).

Table 1: Summary of Performance

Class	PSNR	SSIM	Level	Clsfy Rate
1	27.034	0.645	3	83.825
2	24.618	0.601	2	87.125
3	29.189	0.710	3	86.875
4	31.923	0.827	2	92.45
5	30.508	0.794	3	91.3
6	33.187	0.862	3	93.075

4.1 Effect of the Hierarchical Implementation

One of the novelties of the CFVI is its implementation of a hierarchical architecture that allows coarse reconstruction between the model with the image data using only a small subset of highly salient features, followed iteratively by more finer reconstructions with increased features. The following experiment examines the impact of increasing the levels of decomposition in the hierarchy by comparing with one level decomposition (3 features), two levels of decomposition (4 features) and three levels of decomposition (5 features). The cumulative average performance curves for some of the video clips (Figure 2), suggest that increasing the levels of decomposition, has a significant impact on the performance of motion estimation. In order to prove beyond doubt that the hierarchical implementation with increasing complexity not alone has a positive impact on the motion estimation process but also on correct classification of a a video, we conduct experiments on using different number of features extracted at various

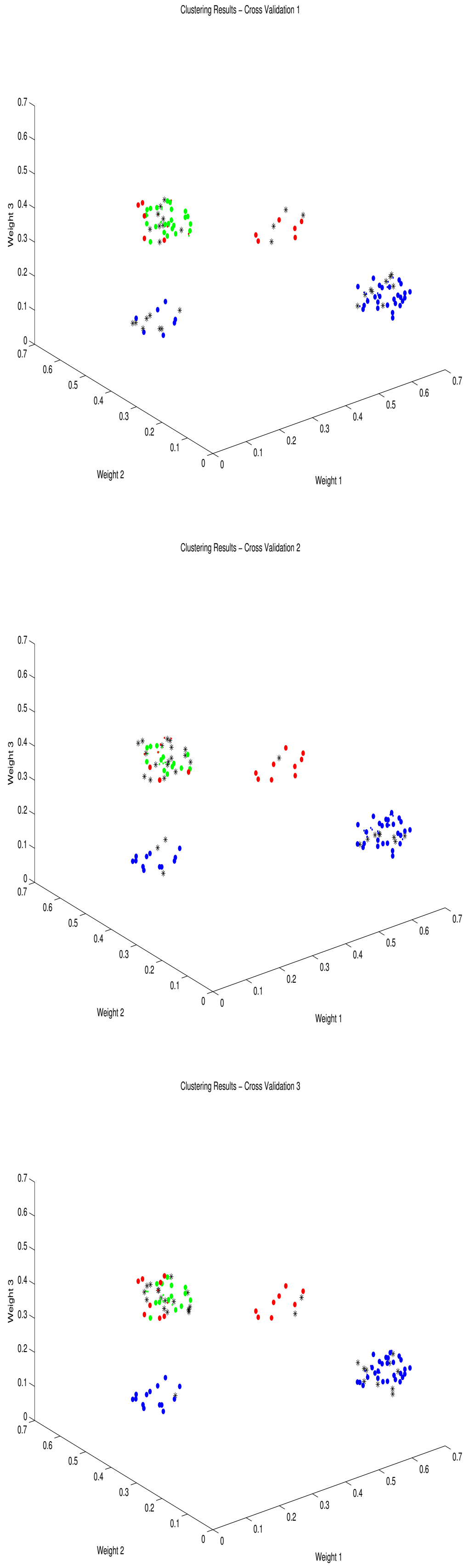
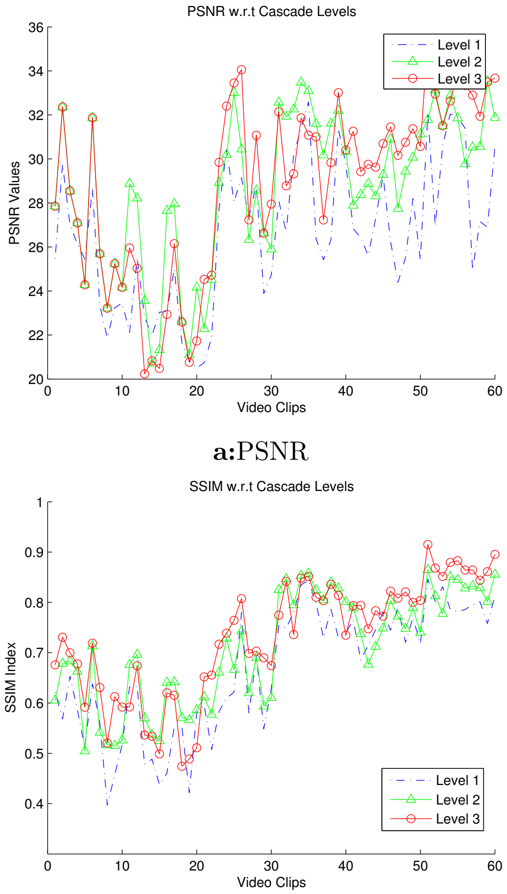


Figure 1: Cross Validation Clustering Results of 3 most salient features (colour, texture and intensity)



a:PSNR

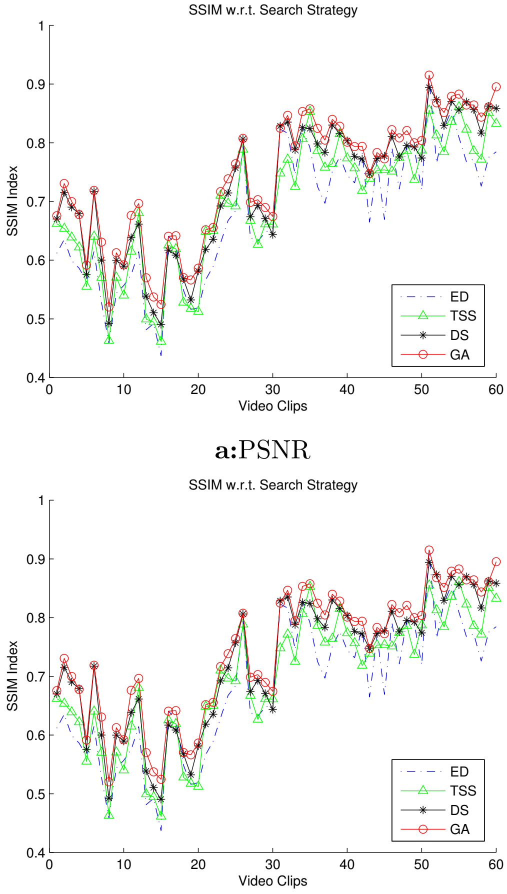
b:SSIM index

Figure 2: Cumulative average performance of motion estimation with respect to number of cascade levels

levels of the hierarchical model and subjecting them to classification. The distribution of the number of features required for correct classification of videos (Figure 4a.), show positive correlation with reference to our observation in Figure 2. However, in some videos smaller number of salient features are found to be more effective than larger subset of unsalient features, mainly due to the ambiguity rise by unsalient feature in the model.

4.2 Effect of search strategy and performance metric

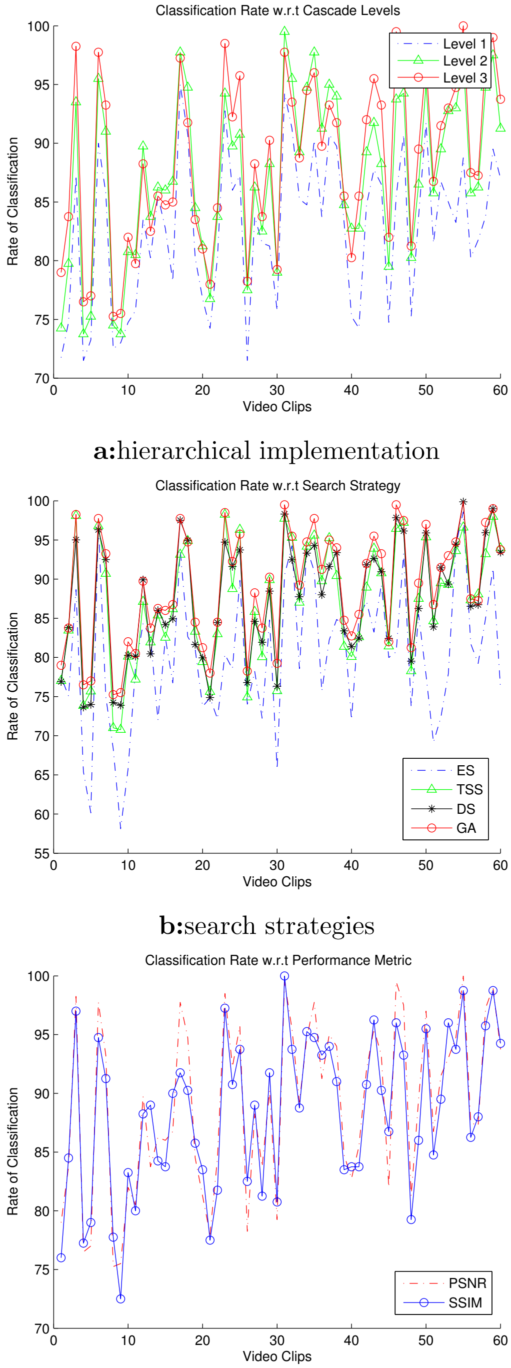
We now conduct experiments that jointly evaluates the effect of different search mechanisms on motion estimation and the use of different performance metrics. We present results in Figure 3 and Figure 4b. showing that our motion estimation model with weighted genetic algorithm search compares favourably with conventional search methods such as diamond search, three step search and exhaustive search that are modified to take weighted features. In addition to the experiments above that already provide sufficient evidence on the ef-



a:PSNR

b:SSIM Index

Figure 3: Cumulative average performance of motion estimation with respect to search strategy



a:hierarchical implementation

b:search strategies

c:performance metric

Figure 4: Classification Rate with respect to: a) number of cascade levels, b) search strategy, c) performance metric

fect of different performance metrics on motion estimation, we also investigated the effect of different performance metrics on the overall classification performance of the model. The curve for each metric in Figure 4c., suggest that the PSNR metric produces maximum rate of classification for a majority of the videos closely followed by the SSIM index and relative entropy.

4.3 Comparison of Video Retrieval

Finally, having investigated on the effect of different system parameters on the model, we compare our video indexing model with simple implementation of a color based video retrieval system. Here, we have built a simplified baseline framework based on the work of [20]. We have performed some initial comparative experiments with both the systems and our results suggest that our proposed methodology increases the classification rate by an average of 8% for all class labels in comparison to the baseline technique. The experiments have also indicated that the proposed strategy is potentially scalable to larger datasets. In terms of the computational demand, the baseline strategy is more efficient (437ms per image frame compared to 783ms for our method) when implemented with Matlab and run on a Pentium Duo Core processor with 4GB RAM. Since there exist a large tradeoff between the efficiency and accuracy of our models, we are working on optimizing our feature level method by incorporating higher level semantic knowledge into indexing.

5 Conclusions

We have investigated a novel framework of combined feature-level motion estimation and classification for low-level video indexing. The hierarchical implementation of the framework with increasing complexity allows greater robustness of the proposed model resulting in a method that outperforms simpler models on standard datasets. One of the open question concerns the optimal set of features that are needed to faithfully describe a video. Initial experimentation suggests that the relationship between a particular video and the necessary feature set required to describe it completely, maybe complex. In our future work, we will address this issue and also look to extend the proposed model to extract higher order semantics in a video to help perform better indexing and retrieval.

References

[1] Caviar test case scenarios, available at http://homepages.inf.ed.ac.uk/rbf/caviardata1/, 2005.

[2] PETS: Performance evaluation of tracking and surveillance, available at http://www.cvg.rdg.ac.uk/slides/pets.html, 2006.

[3] A.Hampapur, R. Jain, and T.E. Weymouth. Feature based digital video indexing. *Proceedings of the third IFIP WG2.6 working conference on Visual database systems 3*, pages 115–141, 1997.

[4] M. Ahmed and A. Karmouch. Video indexing using a high-performance and low-computation color-based opportunistic technique. *Optical Engineering*, 41(2):505–517, Feb. 2002.

[5] F.I. Bashir and A.A. Khokhar. Video content modelling: An overview. *In International Workshop on Frontiers of Information Technology*, 2003.

[6] H. Bhaskar, R.L. Kingsland, and S. Singh. Multiresolution based motion estimation for object tracking using genetic algorithm. *IET Intl. Conf. on Visual Information Engineering*, 1(1):583–588, Sep. 2006.

[7] J. Calic and E. Izquierdo. A multiresolution technique for video indexing and retrieval. *Proceedings of IEEE Intl. Conf. on Image Processing*, 2002.

[8] S-F. Chang, W. Chen, H. Sundaram H.J. Meng, and D. Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *IEEE Transactions on Circuits and Systems for Video Technology,*, 8(5):602–615, Sep. 1998.

[9] J. D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, 30(4):607–625, Apr. 1997.

[10] M. Davis. Knowledge representation for video. *In Working Notes : Workshop on Indexing and Reuse in Multimedia Systems*, pages 19–28, Aug. 1994.

[11] M. Fleischman, H. Evans, and D. Roy. Unsupervised content-based indexing for sports video retrieval. *Proceedings of the 15th Intl. Conf. on Multimedia*, pages 473–474, 2007.

[12] D. Gibson, N. Campbell, and B. Thomas. Visual abstraction of wildlife footage using Gaussian mixture models and the minimum description length criterion. *International Conference on Pattern Recognition*, page 814817, Aug. 2002.

[13] R. Gross and J. Shi. The CMU motion of body (MoBo) database (CMU-RI-TR-01-18), available at http://mocap.cs.cmu.edu/. Technical report, Robotics Inst., Carnegie Mellon Univ, 2001.

[14] R. Lienhart. Comparison of automatic shot boundary detection algorithms. *In Image and Video Processing VII 1999, Proc. SPIE*, 3656(29):290–301, Jan. 1999.

[15] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Video abstracting. *Commununication of ACM*, 40(12):54–62, 1997.

[16] H.R. Naphade and T.S. Huang. A probabilistic framework for semantic video indexing, filtering,and retrieval. *IEEE Transactions on Multimedia*, 3(1):141–151, May. 2001.

[17] S. V. Porter, M. Mirmehdi, and B. T. Thomas. Video indexing using motion estimation. *BMVC*, 1(1):659668, Sep. 2003.

[18] S.V. Porter, M. Mirmehdi, and B.T. Thomas. Detection and classification of shot transitions. *Proceedings of the 12th British Machine Vision Conference*, 1(1):73–82, Sep. 2001.

[19] G-J. Qi, X-S. Hua, Y. Rui, J. Tang, T. Mei, and H-J. Zhang. Correlative multi-label video annotation. *Proceedings of the 15th Intl. Conf. on Multimedia*, 1(1):17–26, 2007.

[20] M. Rautiainen and D. Doermann. Temporal color correlograms for video retrieval. *International Conference on Pattern Recognition*, 1:10267, 2002.

[21] S.W. Smoliar, H.J. Zhang, and J.H. Wu. Using frame technology to manage video. *Proc. of the Workshop on Indexing and Reuse in Multimedia Systems*, B:189194, Nov. 1994.

[22] C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, Jan 2005.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13, 2004.

[24] M-F. Weng and Y-Y. Chuang. Multi-cue fusion for semantic video indexing. *Proceedings of ACM Multimedia*, 1:71–80, Oct. 2008.

[25] J. Yang and A.G. Hauptmann. Exploring temporal consistency for video analysis and retrieval. *Proceedings of the 8th ACM Intl. Workshop on Multimedia Information Retrieval*, 1(1):33–42, 2006.

[26] M. Zampoglou, T. Papadimitriou, and K.I. Diamantaras. Integrating motion and color for content based video classification. *Proceedings of IEEE Intl. Conf. on Image Processing*, 2008.